IN THE CLAIMS:

1. (Currently amended) A method of processing work items in a data processing system comprising the steps of:

generating an interrupt in response to receipt of a work item in the system;

disabling system interrupts pending processing of the work item;

scheduling a task through the generated interrupt for processing of the work item by placing the task on task queue;

executing the task to process the work item when the task reaches the top of the queue;

processing additional work items received by the system; and

when there are no additional work items for processing, speculatively scheduling a further task for processing of subsequently received work items in the system, without enabling system interrupts: interrupts;

executing the speculatively scheduled task to process work items received by the system;

processing one or more work items when at least one work item has been received by the system when the speculatively scheduled task is executed;

repeating the scheduling, executing and processing steps after processing the one or more work items, without enabling system interrupts; and

enabling system interrupts when no additional work items have been received by the system when the speculatively scheduled task is executed.

2. (Canceled)

3. (Previously presented) The method of claim 1, wherein the work items are managed on a queue.

4. (Previously presented) The method of claim 1, wherein the event that further work items are received after the task is scheduled and prior to execution of the task, the step of executing the task comprises processing all the received work items.

5. (Currently amended) A data processing system comprising:

processing means for executing tasks to process work items in the data processing system; and interrupt generating means for generating an interrupt in response to receipt of a work item in the system; wherein the processing means is operable to:

disable system interrupts;

schedule a task through the generated interrupt for processing of the work item;

execute the task to process the work item;

process additional work items received by the system; ~~and~~

when there are no additional work items for processing, speculatively schedule a further task for processing of subsequently received work items in the system, without enabling system ~~interrupts.~~ interrupts;

execute the speculatively scheduled task to process work items received by the system;

process one or more work items when at least one work item has been received by the system when the speculatively scheduled task is executed;

repeat the scheduling, executing and processing steps after processing the one or more work items, without enabling system interrupts; and

enable system interrupts when no additional work items have been received by the system when the speculatively scheduled task is executed.

6. (Canceled)

7. (Previously presented) The data processing system of claim 5, further comprising memory for storing the received work items on a queue.

8. (Previously presented) The data processing system of claim 5, wherein the event that further work items are received after the task is scheduled and prior to execution of the task, the processing means is operable to execute the task to process all the work items.

9. (Previously presented) The data processing system of claim 5, wherein the interrupt

generating means and processing means are embodied in a data storage controller and the work items comprise data transfer requests from an attached host system.

10. (Currently amended) A computer program product comprising a computer usable medium having computer readable program code means embodied in the medium for processing work items in a data processing system, the program code means comprising:

code means for causing the data processing system to generate an interrupt in response to receipt of a work item in the system;

code means for causing the data processing system to disable system interrupts;

code means for causing the data processing system to schedule a task through the generated interrupt for processing of the work item;

code means for causing the data processing system to execute the task to process the work item;

code means for causing the data processing system to process additional work items received by the system; and

code means for causing the data processing system to speculatively schedule a further task for processing of subsequently received work items in the system when there are no additional work items for processing, without enabling system interrupts. interrupts;

code means for causing the data processing system to execute the speculatively scheduled task to process work items received by the system;

code means for causing the data processing system to process one or more work items when at least one work item has been received by the system when the speculatively scheduled task is executed;

code means for causing the data processing system to repeat the scheduling, executing and processing steps after processing the one or more work items, without enabling system interrupts; and

code means for causing the data processing system to enable system interrupts when no additional work items have been received by the system when the speculatively scheduled task is executed.

11. (Canceled)

12. (Currently amended) A method of processing work items in a data processing system, comprising:

effectively providing an interrupt-based mechanism for processing work items, when system utilization is low with respect to work items, wherein system utilization is determined through the speculative scheduling of a task that processes subsequently received work items in the system when there are no additional work items for processing; and

effectively providing a polling-based mechanism for processing work items, when system utilization is relatively high with respect to work items, wherein system utilization is determined through the speculative scheduling of a task that processes subsequently received work items in the system when there are no additional work items for processing.

13. (Previously presented) A method as claimed in claim 12 wherein work items are received in accordance with at least one device driver associated with a host system.

14. (Previously presented) A method as claimed in claim 12 wherein the data processing system comprises a storage controller.